

TP Informatique 06

 On rappelle qu'un script (fichier *.py) doit être enregistré et exécuté (touche F5) pour que les fonctions saisies dans le script soient utilisables dans la console.

Exercice 1

On propose la fonction suivante :

```
def f(n):  
    """f(n:int)->int"""  
    k,res=0,0  
    while k<=n:  
        res+=k  
        k+1  
    return res
```

1. Identifier un défaut majeur dans la fonction. On pourra tester celle-ci.
2. Corriger ce défaut et expliquer le calcul effectué par la fonction `f`.

Exercice 2

On propose la fonction suivante :

```
1 def g(n):  
2     """g(n:int)->int"""  
3     k,res=1,1  
4     while k<=n:  
5         res*=k  
6         k+=1  
7     return res
```

1. Que renvoie la fonction `g(n)` d'argument `n` un entier ?
2. On échange les lignes 5 et 6 de la fonction.
 - (a) Que renvoie la nouvelle version de la fonction ?
 - (b) Modifier la condition de la boucle pour que la fonction renvoie le même résultat que précédemment.

Exercice 3

1. Écrire une fonction `somme1(n)` avec boucle conditionnelle d'argument `n` un entier qui renvoie $\sum_{0 \leq 2k+1 \leq n} (2k+1)^2$.
2. Écrire une fonction `somme2(n)` avec boucle inconditionnelle d'argument `n` un entier qui renvoie $\sum_{0 \leq 2k+1 \leq n} (2k+1)^2$.

On vérifiera sur différentes valeurs de n que les fonctions `somme1` et `somme2` renvoient les mêmes résultats.

Exercice 4

Pour chacune des fonctions suivantes, on veillera à calculer efficacement les puissances successives de q^k .

1. Écrire une fonction `geom1(n,q)` avec boucle inconditionnelle d'arguments `n` un entier et `q` un flottant qui renvoie la valeur de $\sum_{k=0}^n q^k$.
2. Écrire une fonction `geom2(n,q)` avec boucle conditionnelle d'arguments `n` un entier et `q` un flottant qui renvoie la valeur de $\sum_{k=0}^n q^k$.

Exercice 5

On a
$$\sum_{k=0}^n k = \frac{n(n+1)}{2} \xrightarrow{n \rightarrow \infty} +\infty$$

Par conséquent, on peut définir

$$\forall p \in \mathbb{R} \quad \varphi(p) = \min \left\{ n \in \mathbb{N} \mid \sum_{k=0}^n k \geq p \right\}$$

Écrire une fonction `seuil(p)` d'argument `p` un flottant et qui renvoie le couple $\left(\sum_{k=0}^{\varphi(p)} k, \varphi(p) \right)$.

Tester la fonction pour différentes valeurs de `p`.

Exercice 6

Dans l'éditeur, effectuer l'importation `import numpy as np`.

1. Écrire une fonction `somme_log(n)`, avec boucle conditionnelle, d'argument `n` un entier et qui renvoie la valeur de $\sum_{1 \leq k, k \ln(k) \leq n} k$. On rappelle que la fonction `ln` correspond à la fonction `log` dans `numpy`.
2. Peut-on écrire une version avec boucle inconditionnelle ? Si oui, proposer une telle version.

Exercice 7

Soit n entier non nul et p un nombre premier. On appelle *p-valuation* de n l'entier noté $v_p(n)$ défini par

$$v_p(n) = \max \{ k \in \mathbb{N} \mid p^k \text{ divise } n \}$$

1. Écrire une fonction `val(n,p)` d'arguments `n` un entier non nul, `p` un nombre premier qui renvoie la p -valuation de n .
2. Identifier le cas le plus défavorable, *i.e.* celui qui demande le plus de calculs, pour la fonction `val`.